

## Занятие 11: Java EE, первая часть



Главное отличие Enterprise-разработки это ориентация на бизнес. Как правило, если люди говорят об Enterprise-приложениях, они имеют в виду следующие аспекты:

- ✓ Сложные проекты
- ✓ Высокое качество кода
- ✓ Иерархия в коллективе разработчиков
- ✓ Высокая степень ответственности
- ✓ Ориентирование на результат
- ✓ Работа в терминах бизнеса

- ✓ Надежность
- ✓ Скорость разработки
- ✓ Время жизни проекта
- ✓ Преемственность
- ✓ Много разработчиков

# Концепции Enterprise-приложений

- ✓ **IoC(Inversion of Control)**
- ✓ **DI(Dependency Injection)**
- ✓ **IoC-container**

Все эти концепции нацелены на то, чтобы снизить связанность компонентов системы, чтобы изменения одного компонента не вызывали каскадные изменения остальных.

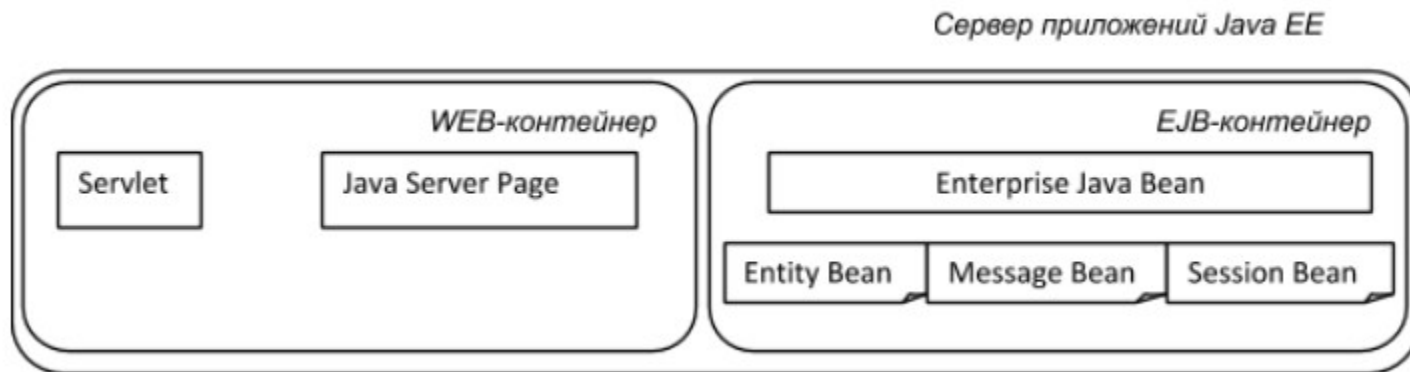
Особенность enterprise-приложений состоит в том, что все, или почти все, из них основаны на этих принципах.

# Особенность Enterprise-приложений

Технологии, включенные в Java EE, не могут быть запущены просто в JVM. Они работают внутри web- и EJB-контейнеров.

Для запуска enterprise-приложений существуют сервера приложений, которые реализуют Java EE спецификацию.

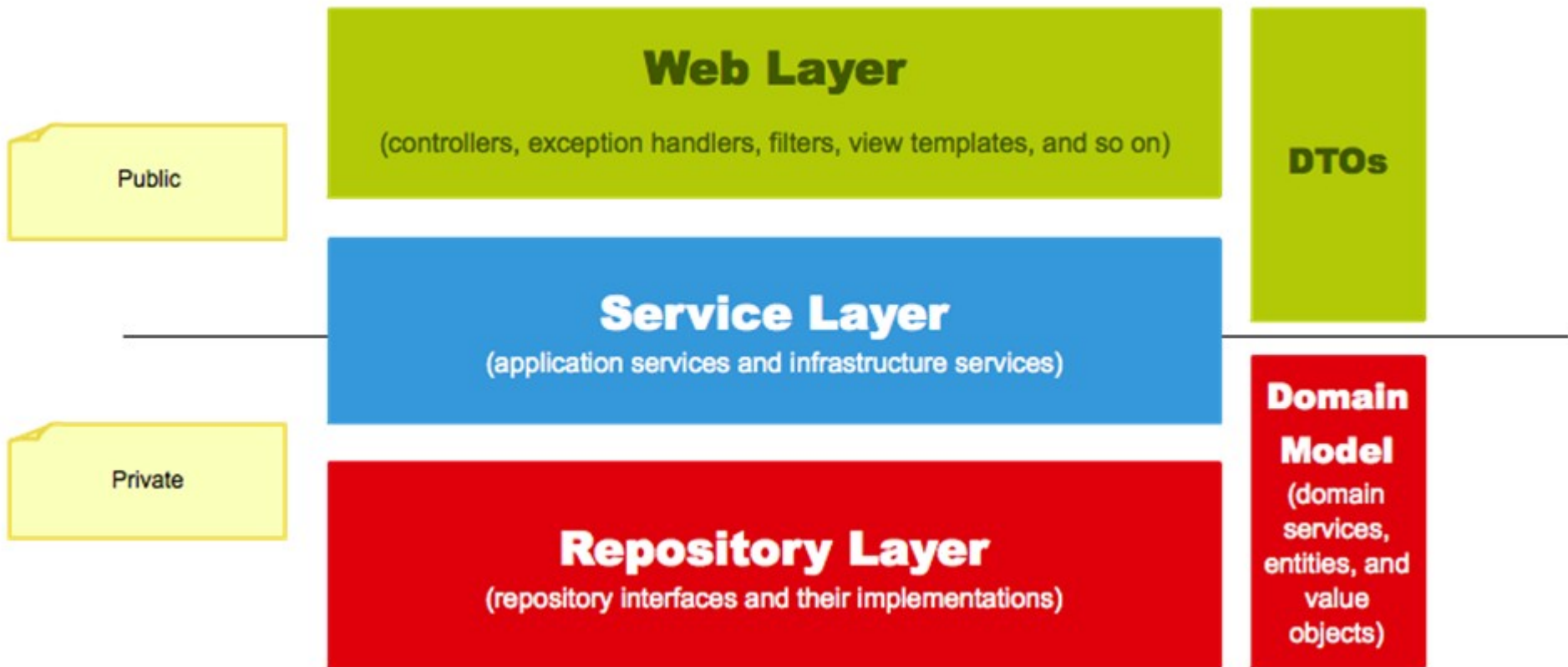
# Состав сервера приложений



# Состав Java EE



# Слоистая архитектура





- ✓ Servlet API - одна из ключевых технологий в EE. Отвечает за обработку запросов от пользователей к веб-серверу.
- ✓ JSP (Java Server Pages с расширением JSTL) - отображение веб-страниц. Простая возможность генерации веб-страниц из Java
- ✓ Java Server Faces - создает JS-агента, жестко связанного с бэкендом
- ✓ GWT - google web toolkit, генератор веб+js-страниц, осуществляет связь между ними на стороне сервера.

# Сервлеты

Сервлет – класс, который предназначен для обработки пользовательских запросов, приходящих на определенный URL веб-сервера.

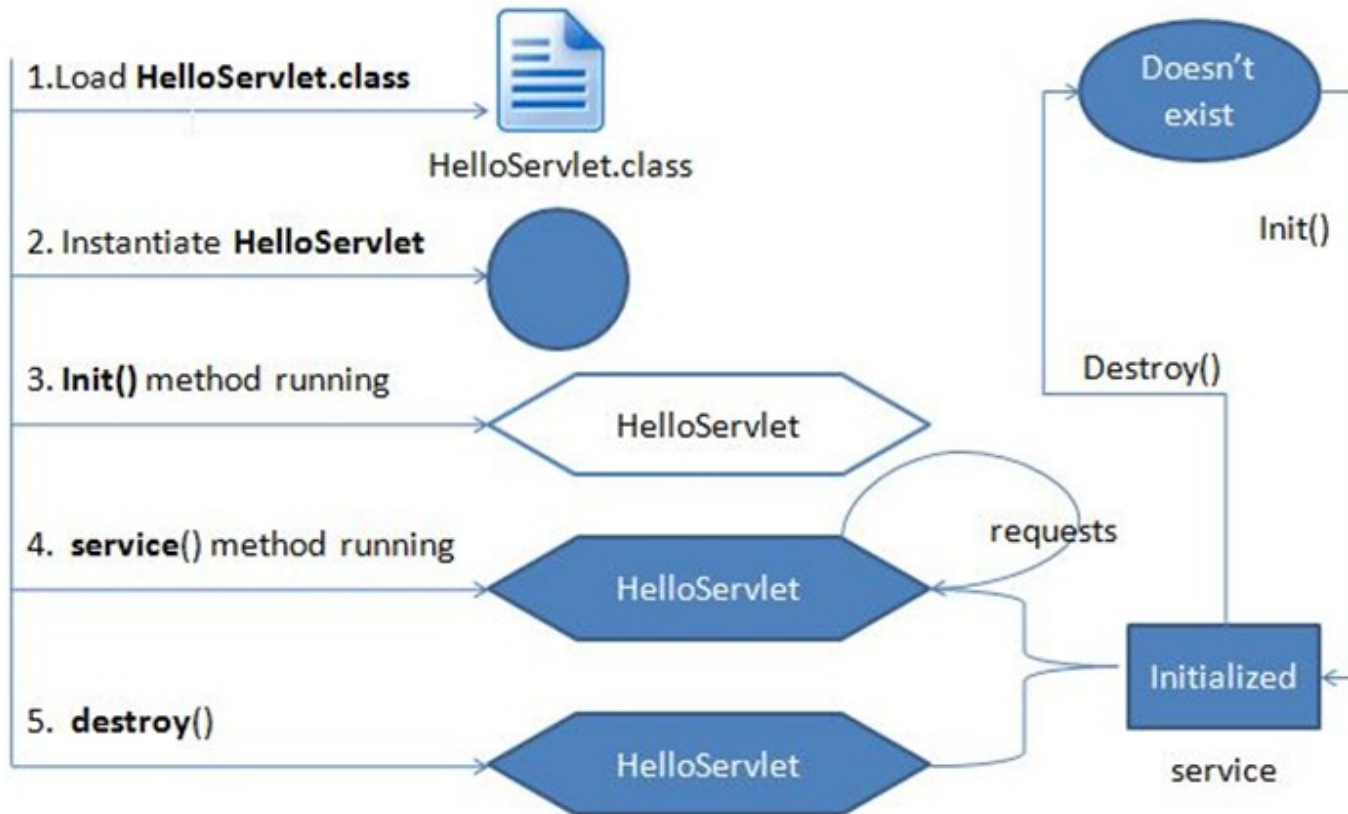
Программа состоит из нескольких классов-сервлетов.

Сервлеты как правило обрабатывают GET и POST запросы.

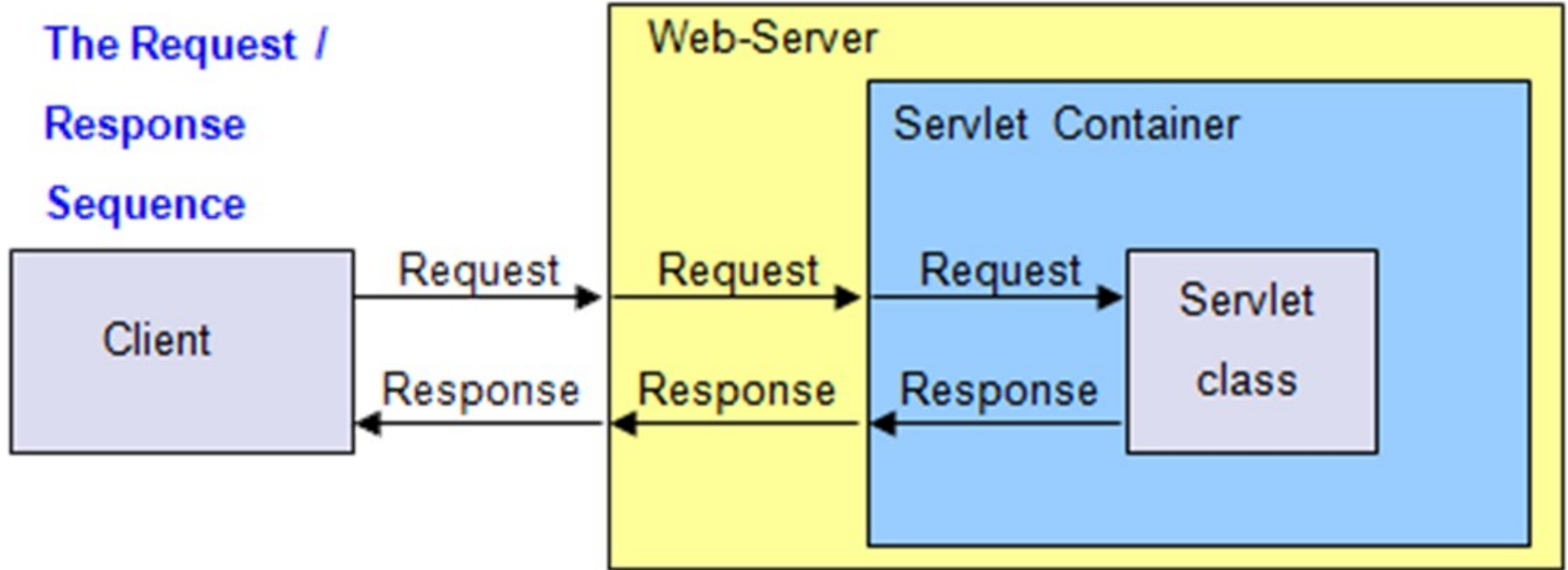
При старте приложение находит все классы сервлетов и привязывает каждый к определенному URL.

Контейнер сервлетов – хранилище экземпляров сервлетов. Объектный пул, который предоставляет пользователю конкретный экземпляр для работы.

# Жизненный цикл сервлетов



# Контейнер сервлетов



# Концепция бина (bean)

Bean – боб, зерно. Это некая единица бизнес-логики. По спецификации EJB бины бывают нескольких видов:

- ✓ Сущности (entity)
- ✓ Менеджеры (операции, сервисы), агрегаторы бизнес-логики

Изначально концепция предполагала только наличие сущностей.

Spring ввел возможность аспектно-ориентированного программирования. Со временем EJB и Spring интегрировались друг в друга. EJB стал использовать IoC, Spring стал использовать бины.

# Концепции построения web-сервисов

Основное отличие Rest от Soap. Rest это принципы, Restful – сервис, поддерживающий все принципы rest (возможно только для сервисов, не имеющих состояний). Soap – протокол.

Все протоколы делятся на

- ✓ Soap-протоколы
- ✓ Не-soap протоколы

Активно используются следующие технологии:

- ✓ Jax-WS (web services)
- ✓ Jax-RS (RESTful services)

Спецификация для создания масштабируемых клиентов для работы с БД. Удобно для баз, в которых существует большое количество сущностей, они постоянно меняются, между ними существует большое количество связей.

Самая популярная реализация спецификации JPA – **Hibernate**

Помимо JPA он реализует Java Cache API

Веб-сервисов недостаточно для обмена данными между несколькими приложениями. На высоконагруженных приложениях веб-сервисам не хватает производительности

JMS – спецификация для реализации асинхронных сообщений.

Службы обмена сообщениями, основанные на JMS, позволяют реализовывать высокопроизводительные службы обмена сообщениями, основанные на очередях и гарантирующие (почти) доставку.

Существует большое количество систем, решающих проблему асинхронного обмена сообщениями (Rabbit MQ, Kafka) без поддержки JMS



# Что нужно знать (минимум)

- ✓ Java Core
- ✓ Concurrency
- ✓ Collections
- ✓ Spring (core, MVC, Security)
- ✓ JMS
- ✓ Jax-ws
- ✓ Jax-rs
- ✓ Hibernate (JPA)

- ✓ Многопоточность сервлетов – из коробки. Следовательно, слабо кастомизируется
- ✓ При использовании состояний в сервлете, необходимо очищать их после использования
- ✓ Не нужно хранить ссылки на сервлеты (не будут очищаться)
- ✓ Желательно правильно распределять сервлеты по урлам